CMOS Fuzzy Decision Diagram Implementation

V. VARSHAVSKY [*], I. LEVIN[*], V. MARAKHOVSKY[†],
A. RUDERMAN[*], N. KRAVCHENKO[‡]
[*]School of Engineering, Bar Ilan University, ISRAEL
[†] The University of Aizu, JAPAN
[‡]Advanced Logic Projects Ltd., ISRAEL

Abstract: - The subject of the study is design of multi-valued (analog) CMOS fuzzy controllers. A functional completeness of summing amplifier with saturation in a multi-valued logic of an arbitrary value proven in previous works gives a theoretical background for analog implementation of fuzzy devices. Compared with the traditional approach based on explicit fuzzification / defuzzification procedures analog fuzzy implementation has the advantages of higher speed, lower power consumption, smaller die area and more. In the present paper, we expand functional capabilities of summing amplifier by using "masking of the input". The paper provides design example for an industrial fuzzy controller implementation by the proposed mask circuit and SPICE simulations of the controller.

Key-Words: Fuzzy Logic, Multi-Valued Logic, Decision Diagrams, Fuzzy Controller, CMOS Circuits.

1 Introduction

We showed in [1, 2] that a summing amplifier with saturation is a functionally complete element in any multi-valued logics (of an arbitrary value), and that it may serve a basis for implementing analog fuzzy devices.

In this paper, we expand functional capabilities of such an amplifier by using a minor complication of the device that is called "masking of the input".

The subject of the study is design of multi-valued (analog) CMOS circuits that present functions implementable by fuzzy controllers.

Without loss of generality let's suppose the odd value logic ($k = 2a + 1$ and $-a \le x_j \le +a$).

A multi-valued logic function describing behavior of fuzzy controllers can be specified by two adequate methods: by a system of fuzzy rules applied to linguistic variables of the type " if A then B" or by a table where a set of linguistic variables corresponds to a value of an output linguistic variable.

Let us assume that $X = \{x_0, x_1, \cdots, x_{n-1}\}$ - is a set of input multi-valued variables and $y = F(X)$ - is an output variable, then for a multi-valued logic function we may build an analog of the Shannon's decomposition in the binary logic:

$$y_i = F_i(X) = \bigcup_{\alpha=-a}^{+a} [\textbf{if } x_j = \alpha \textbf{ then } y = F(x_j = \alpha, X \setminus x_j)]. \quad (1)$$

Equation (1) can be considered as a multi-valued analog of Binary Decision Diagram (BDD). It can be further expanded enabling building a fuzzy circuit by using variables expulsion method. For this purpose, we need a sub-circuit implementing the function:

$$\textbf{if } Z = A \quad \textbf{then } y = F(Z = A, X \setminus Z)\ ^{1} \quad (2)$$

where $Z \subset X$ and A is a value combination of variables Z. It is naturally to call diagrams received as a result of sequential variables expulsion according to (1) as Fuzzy Decision Diagrams (FDD).

[1] **else** in (2) will be defined by circuit requirements.

Having a basic element (sub-circuit) realizing (2), we can implement a fuzzy device directly according to the system of fuzzy rules. We consider a resulting device as an implementation of the FDD. However, note that equations (1) and (2) represent multi-valued functions in a piece-wise constant manner. An example of a 7-valued function is given in Fig.1. The function of Fig.1,a satisfies the rule table (matrix) but does not account for fuzzification and defuzzification procedures. Accounting for these procedures, the fuzzy function should be piece-wise linear. Fig.1,b gives an example of such a representation having evenly distributed input and output linguistic variables in the functions of fuzzification and defuzzification.

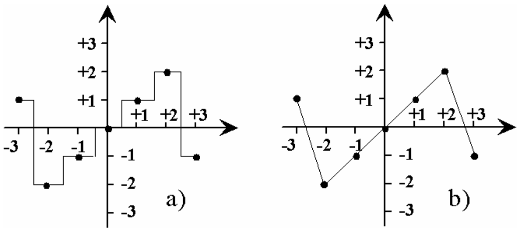

Figure 1. 7-valued function of one variable: piece-wise constant representation (a), piece-wise linear representation (b).

2 Masking Summing Amplifier Input

As in [1,2], we will consider an inverting summing amplifier

$$S(A\cdot X;\beta)=\begin{cases} +a & \textbf{if} \quad \sum_{i=1}^{n}\alpha_i\cdot x_i+\beta\leq -a \\ -\sum_{i=1}^{n}\alpha_i\cdot x_i-\beta & \textbf{if} \quad +a>\sum_{i=1}^{n}\alpha_i\cdot x_i+\beta>-a \\ -a & \textbf{if} \quad +a\leq\sum_{i=1}^{n}\alpha_i\cdot x_i+\beta \end{cases} \qquad (3)$$

where $A=\{\alpha_1,\alpha_2,...,\alpha_n\}$ – weight coefficients, $X=\{x_1,x_2,...,x_n\}$ – analog or multi-valued variables, β – offset (shift), $\pm a$ – saturation values (in the case of k-valued logic, $k=2a+1$).

Let us introduce a masking function $M_\alpha(x)$ such that

$$M_\alpha(x)=\begin{cases} +a & \textbf{if} \quad x\leq\alpha-1 \\ a(\alpha-x) & \textbf{if} \quad \alpha-1<x<\alpha+1 \\ -a & \textbf{if} \quad \alpha+1<x \end{cases} \quad (4)$$

where $-a\leq\alpha\leq a$ is a fixed value of the variable x. It can be easily seen that as $x=\alpha$ $M_\alpha(\alpha)=0$. Fig. 2 illustrates an example of the function $M_{-1}(x)$ for $k=7$.

Taking into account that the source voltage V_{dd} has the logical value $+a$ and the ground potential V_{gnd} has the logical value $-a$, the mask-function can be easily implemented on the base of summing amplifier as

$$M_\alpha(x)=\begin{cases} S(ax;|\alpha|V_{dd}) & \textbf{if} \quad \alpha<0 \\ S(ax) & \textbf{if} \quad \alpha=0 \\ S(ax;|\alpha|V_{gnd}) & \textbf{if} \quad \alpha>0 \end{cases} . \quad (5)$$

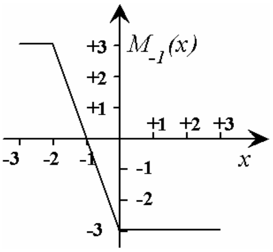

Figure 2. $M_{-1}(x)$ diagram for $k=7$.

Using the mask-function it is possible to implement the fuzzy rule

$$\textbf{if } x=\alpha \textbf{ then } y=F(x=\alpha,Y) \textbf{ else } y=0, \quad (6)$$

that extracts a value of the function $F(x=\alpha,Y)$ in the point $x=\alpha$, as the circuit built of summing amplifiers shown in Fig.3.

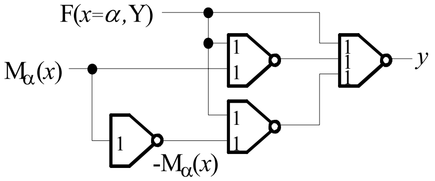

Figure 3. Implementation of the rule (6).

This implementation can be written in analytical form as

$$y = S(S(M_\alpha(x); F(x=\alpha, Y)); S(-M_\alpha(x); F(x=\alpha, Y)); F(x=\alpha, Y)). \tag{7}$$

For example, if $\alpha = -1$, $F(x=-1, Y) = 2$, and k = 7, the behavior of the circuit in Fig.3 can be illustrated by Fig.4.

The wider variable change range

$$\text{"}\mathbf{if}\ \ \gamma \le x \le \delta\ \ \mathbf{then}\quad y = F(\gamma \le x \le \delta, Y) = \Phi(Y)\ \ \mathbf{else}\ \ y = 0\text{"} \tag{8}$$

study is shown in Fig. 5.

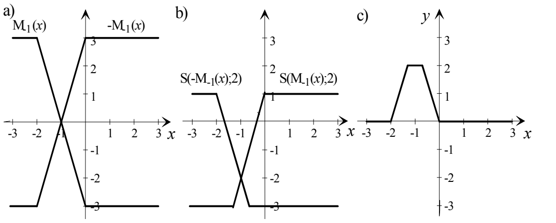

Figure 4. Example of implementing the rule " if *x=-1* then *y=2* else *y=0* ", (*k*=7).

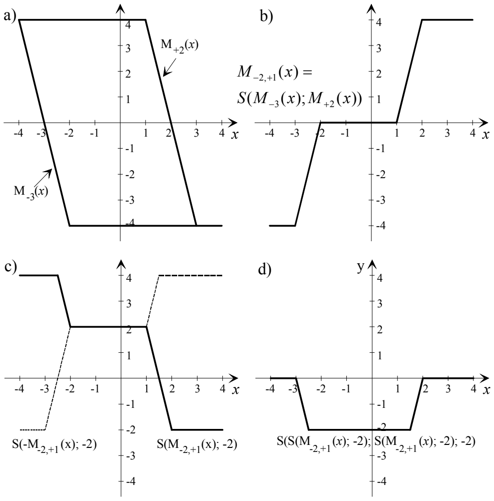

Figure 5. Example of building a mask for the condition (k = 9): " **if** $-2 \le x \le +1$ **then** $y = -2$ **else** $y = 0$".

Similar to the order of operations performed when building diagrams of Fig.5, the rule (8) can be implemented as

$$\begin{cases} M_{\lambda,\delta}(x) = S(M_{\lambda-1}(x); M_{\delta+1}(x)); \\ -M_{\lambda,\delta}(x) = S(M_{\lambda,\delta}(x)); \\ y = S(S(M_{\lambda,\delta}(x); \Phi(Y)); \\ S(-M_{\lambda,\delta}(x); \Phi(Y)); \Phi(Y)). \end{cases} \tag{9}$$

This implementation corresponds to the circuit presented in Fig.3 if the inputs $M_\alpha(x)$ and $F(x=\alpha, Y)$ are changed to $M_{\lambda,\delta}(x)$ and $\Phi(Y)$ respectively.

3 Example

To further illustrate the matter discussed above, let us recall an example from *"A Fuzzy Logic Force Controller for a Stepper Motor Robot", J.G. Hollinger, R.A. Bergstrom, and J.S. Bay, Bradley Department of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.*

Fuzzy controller implements the function of two linguistic variables - "position error" and "force error" - that will be designated as x and y respectively. The variables x and y each have 7 values NL, NM, NS, ZE, PS, PM, PL and their membership functions are shown in Fig.6.

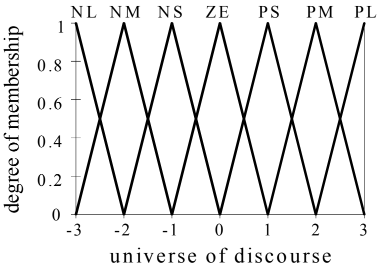

Figure 6. Fuzzy sets for force error and position error inputs.

The Inference Engine Rule Matrix for the output linguistic variable from the cited work is shown in Table 1.

Table 1. Inference Engine Rule Matrix

		position error (x)						
		NL	NM	NS	ZE	PS	PM	PL
force error (y)	NL	NM	NL	NL	NL	NL	NL	NM
	NM	NS	NM	NM	NM	NM	NM	NS
	NS	ZE	NS	NS	NS	NS	NS	ZE
	ZE	ZE	ZE	ZE	ZE	ZE	ZE	ZE
	PS	ZE	PS	PS	PS	PS	PS	ZE
	PM	PS	PM	PM	PM	PM	PM	PS
	PL	PM	PL	PL	PL	PL	PL	PM

Let us transform the Table 1 into the Table 2 to generate fuzzy inference calculating values of corresponding multi-valued logic function.

Table 2. Inference engine rule matrix as the multi-valued logic function

		position error (x)						
		-3	-2	-1	0	1	2	3
force error (y)	-3	-2	-3	-3	-3	-3	-3	-2
	-2	-1	-2	-2	-2	-2	-2	-1
	-1	0	-1	-1	-1	-1	-1	0
	0	0	0	0	0	0	0	0
	1	0	+1	+1	+1	+1	+1	0
	2	1	2	2	2	2	2	1
	3	2	3	3	3	3	3	3

Table 2 comprises only two different columns defining two force error functions (Table 3).

Table 3. Two different functions of the force error

	force error (y)						
	-3	-2	-1	0	1	2	3
$F_1(y)$	-2	-1	0	0	0	1	3
$F_2(y)$	-3	-2	-1	0	1	2	3

Fig.7 shows the graphs of these functions.

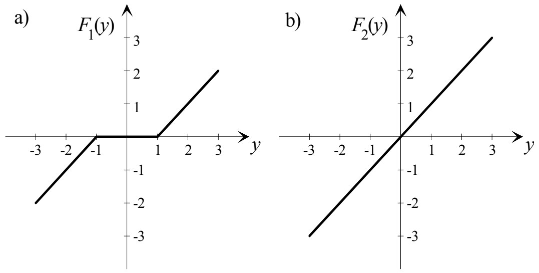

Figure 7. Components of the function decomposition defined by Table 1 for *x*.

Similar to (4), (5), Fig.5,a, and Fig.5,b, it is possible to construct the function $F_1(y)$ in accordance with the sequence of graphs in Fig.8. As a result, the functions $F_1(y)$ and $F_2(y)$ can be implemented as

$$F_1(y) = S_3(\tfrac{2}{3}S_2(\tfrac{3}{2}y;\tfrac{3}{2}V_{gnd});\tfrac{2}{3}S_1(\tfrac{3}{2}y;\tfrac{3}{2}V_{dd}));$$
$$F_2(y) = y\,. \qquad (10)$$

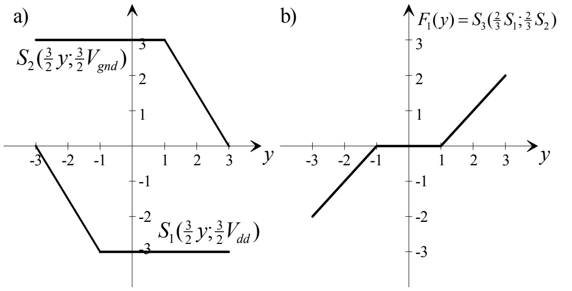

Figure 8. Constructing the function $F_1(y)$.

It can be seen from the Table 1 and Table 2 that the behavior of the controller's output decomposed by variable *x* has the form:

if $NM \le x \le PM$ **then** $Output = F_2(y)$

else $Output = F_1(y)$

or

$$\begin{array}{l} \textbf{if } -2 \le x \le +2 \textbf{ then } Output = F_2(y) \\ \textbf{else } Output = F_1(y) \end{array} \quad . \quad (11)$$

It is possible to split the rule (11) into two rules:

$$\begin{array}{l} \text{if } -2 \le x \le +2 \text{ then } Output = F_1(y) \\ \text{else } Output = 0 \end{array} \quad , \quad (12)$$

$$\begin{array}{l} \text{if } -2 \le x \le +2 \text{ then } Output = 0 \\ \text{else } Output = F_2(y) \end{array} \quad . \quad (13)$$

The rule (12) can be implemented in accordance with (9), (10), and (7) as

$$\begin{cases} M_{-2,+2}(x) = S_4(M_{-3}(x); M_{+3}(x)); \\ -M_{-2,+2}(x) = S_5(M_{-2,+2}(x)); \\ \Phi_1 = S_{11}(S_6(M_{-2,+2}(x); F_1(y)); \\ S_7(-M_{-2,+2}(x); F_1(y)); F_1(y)). \end{cases}$$

To implement the rule (13) let us first implement the rule (14):

$$\begin{array}{l} \text{if } -2 \le x \le +2 \text{ then } Output = F_2(y) \\ \text{else } Output = 0 \end{array} \quad (14)$$

as

$$S_{10}(S_8(M_{-2,+2}(x);F_2(y));S_9(-M_{-2,+2}(x); F_2(y));F_2(y)) \,.$$

The rule (13) can be represented as

if $-2 \le x \le +2$ then

$Output = F_2(y) - F_2(y)$ else $Output = 0 + F_2(y)$

with implementation

$$\Phi_2 = S_{13}(S_{12}(S_{10}); F_2(y))\,.$$

Finally, the controller output can be calculated as

$$Output = S_{15}(S_{14}(\Phi_1;\Phi_2)) .$$

It is possible to simplify the suggested controller implementation. Since the summing amplifier S_{10} works without saturation, let us implement the function $-\Phi_2$ as

$$-\Phi_2 = S_{10}(S_8;F_2(y));S_9;F_2(y));2F_2(y)) .$$

Then the controller implementation expression will look like

$$Output{=}\ S(-\Phi_1;-\Phi_2) = S_{11}(S_6;S_7;F_1(y);S_{10})$$

Fig.9 illustrates the structural scheme of the controller implementation (the input weights are marked).

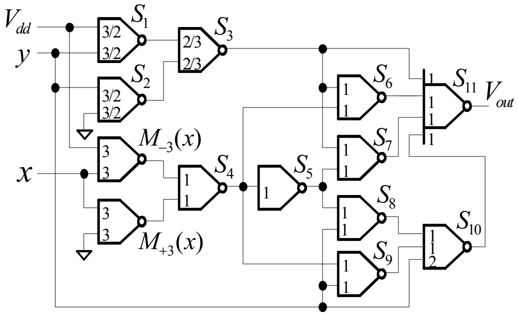

Figure 9. Structural diagram of the controller.

4 SPICE Simulation Results

We will use three-stage and five-stage push-pull CMOS operational amplifiers with 2 MOhm feedback resistors as a basic building block for the controller represented in Fig.9. Examples of such summing amplifiers are shown in Fig.10.

The controller circuit for experiments is given in Fig.11. Functioning of the controller has been checked with SPICE simulation (MSIM 8). MOSIS BSIM3v3.1, 7 level models of 0.4μm transistors have been used.

The source voltage in our experiments was 3.5V. The variable x changed linearly from 0V to 3.5V.

The variable y changed discretely in accordance with its logical values and was kept constant within one cycle of x change.

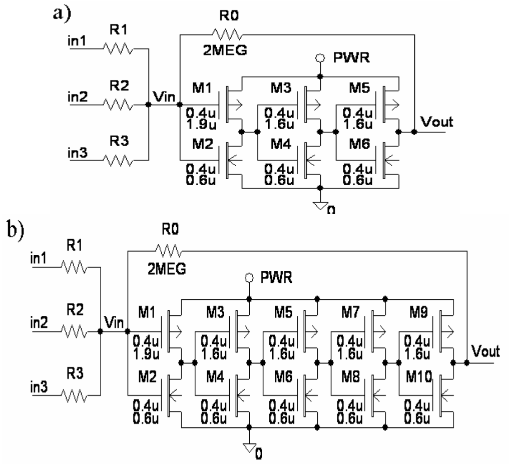

Figure 10. Implementation of the summing amplifier using the push-pull operational amplifier with 3 (a) and 5 stages (b).

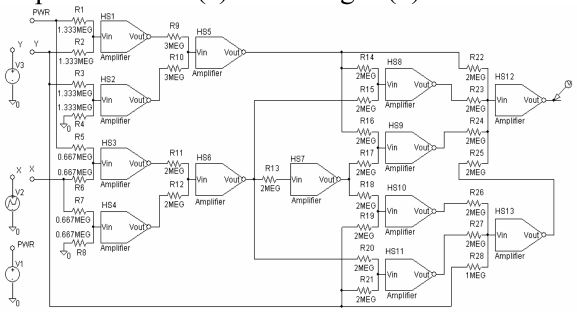

Figure 11. The controller circuit for experiments.

SPICE simulation results confirm the controller correct logical functioning. The accuracy of output signal voltage levels is not so good for controller design based on three-stage amplifiers and becomes much better if five-stage amplifiers are used. This allows increasing considerably the number of logical gradations (e.g., 11 – 13 instead of 7).

For the controller constructed of 3-stage elements SPICE simulation results are shown in Fig.12. It is seen that the controller functioning is correct - logical values of the circuit output depend on the logical values of the input variables in accordance with the Table 2. The accuracy of output signal voltage levels correspondence to its logical levels is not so good. It becomes much better if 3-stage amplifiers are replaced by 5-stage ones. SPICE simulation results for this case are given in Fig.13. Using 5-stage amplifiers it is possible to considerably increase the number of logical gradations (e.g., 11 – 13 instead of 7).

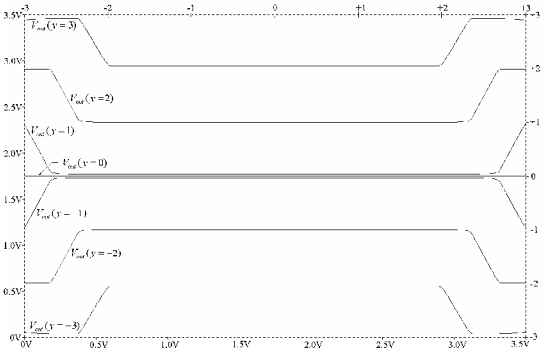

Figure 12. SPICE simulation results for the controller constructed from 3-stage summing amplifiers.

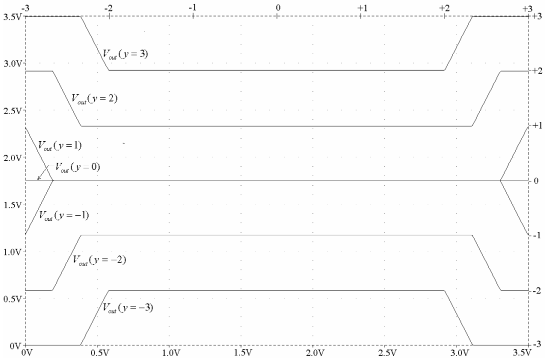

Figure 13. SPICE simulation results for the controller constructed from 5-stage summing amplifiers.

5 Conclusions

Analog implementation of fuzzy controllers has the advantages of better speed and reliability, lower power consumption etc. Theoretically, it is possible due to a functional completeness of summing amplifier with saturation in a multi-valued logic of an arbitrary value. In this paper, suggested is an "input masking" procedure that may be a useful instrument for a systematic analog fuzzy controllers design. To solidify our theoretical considerations, we provided a real life stepper motor fuzzy controller design example and performed SPICE simulations.

References:

[1] V. Varshavsky, V. Marakhovsky, I. Levin, and N. Kravchenko, Fuzzy Device, New Japanese *Patent Application No. 2003-190073, filed to Japan's Patent Office*, Feb. 6, 2003.

[2] V. Varshavsky, V. Marakhovsky, I. Levin, and N. Kravchenko, Summing Amplifier as a Multi-Valued Logical Element For Fuzzy Control, *WSEAS Transactions on Circuit and Systems*, Issue 3, Volume 2, July 2003, pp. 625 – 631.

[3] J. G. Hollinger, R. A. Bergstrom and J. S. Bay. A Fuzzy Logic Force Controller for a Stepper Motor Robot, in: *Fuzzy logic technology and applications* Robert J. Marks II, editor; IEEE Technical Activities Board.